

SQL Cheat Sheet: Intermediate - LIKE, ORDER BY, GROUP BY



| Command | Syntax | Description | Example |
|----------|---|--|---|
| LIKE | <code>SELECT column1, column2, ... FROM table_name WHERE columnN LIKE pattern;</code> | <code>LIKE</code> operator is used in a <code>WHERE</code> clause to search for a specified pattern in a column. There are two wildcards often used in conjunction with the <code>LIKE</code> operator which are percent sign(<code>%</code>) and underscore sign (<code>_</code>). | <code>SELECT f_name , l_name FROM employees WHERE address LIKE '%Elgin, IL%';</code> |
| BETWEEN | <code>SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2;</code> | The <code>BETWEEN</code> operator selects values within a given range. The values can be numbers, text, or dates. The <code>BETWEEN</code> operator is inclusive: begin and end values are included. | <code>SELECT * FROM employees WHERE salary BETWEEN 40000 AND 80000;</code> |
| ORDER BY | <code>SELECT column1, column2, ... FROM table_name ORDER BY column1, column2, ... ASC DESC;</code> | <code>ORDER BY</code> keyword is used to sort the result-set in ascending or descending order. The default is ascending. | <code>SELECT f_name, l_name, dep_id FROM employees ORDER BY dep_id DESC, l_name;</code> |
| GROUP BY | <code>SELECT column_name(s) FROM table_name WHERE condition GROUP BY column_name(s) ORDER BY column_name(s);</code> | <code>GROUP BY</code> clause is used in collaboration with the <code>SELECT</code> statement to arrange identical data into groups. | <code>SELECT dep_id, COUNT(*) FROM employees GROUP BY dep_id;</code> |

Author(s)

[Lakshmi Holla](#)

Changelog

| Date | Version | Changed by | Change Description |
|------------|---------|---------------|--------------------|
| 2021-07-28 | 1.0 | Lakshmi Holla | Initial Version |